

KOMPRESI CITRA BERWARNA DENGAN OBDD

Handoko, Donny KS, Victor G.U.

Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana Salatiga

e-mail: handoko90@yahoo.com,

ABSTRAK: *Ordered Binary Decision Diagram (OBDD)* telah digunakan untuk mengurangi penyimpanan dan waktu perhitungan yang dibutuhkan untuk menguji kebenaran untai digital. OBDD juga telah digunakan sebagai algoritma kompresi citra grayscale dengan memandang sebuah citra sebagai fungsi Boolean dengan *karnaugh-map*. Pada makalah ini OBDD diperluas penggunaannya untuk kompresi citra berwarna. Ada dua mode kompresi yang dilakukan, yaitu *lossless* dan *lossy*. Pada mode *lossy*, digunakan pengubahan RGB ke YUV sebelum OBDD diterapkan. Kompresi OBDD *lossless* mencapai rasio kompresi 1.2, 63.3 dan 1.3 untuk kelompok citra natural dan tekstur. Hasil ini lebih buruk dibandingkan dengan *Lossless JPEG*. Untuk mode *lossy* dicapai rasio kompresi sebesar 2.5, 92.0, 2.6 dan 13.5 untuk citra natural, tekstur dan teks. Hasil ini lebih baik daripada JPEG2000 untuk kategori tekstur dan teks. Penilaian secara subyektif kompresi OBDD dan YUV 4:1:1 sama baik dengan JPEG2000 pada kategori natural dan tekstur tapi lebih buruk pada kategori teks.

Kata kunci: OBDD, YUV, RGB, kompresi citra, *lossless*, *lossy*.

ABSTRACT: *Ordered Binary Decision Diagram (OBDD)* has been used to reduce the amount of space and computing time required for verifying digital circuits. OBDD has also been used to compress grayscale images in form of Boolean function and modeled as a *karnaugh-map*. In this paper, OBDD is used to compress color image as *lossless* (OBDD alone) and *lossy* (YUV-OBDD). *Lossless OBDD* reaches compression ratio 1.2; 63.3 and 1.3 for natural and texture images which are worse compare to *Lossless JPEG*. *Lossy OBDD* reaches compression ratio 2.5; 92.0, 2.6 and 13.5 for natural, texture and text images which are better than JPEG2000 for texture and text images. Subjectively, OBDD combined with YUV compression has the same quality result as JPEG2000 in natural and texture images but worse for text image.

Keywords: OBDD, YUV, RGB, Image Compression, *lossless*, *lossy*.

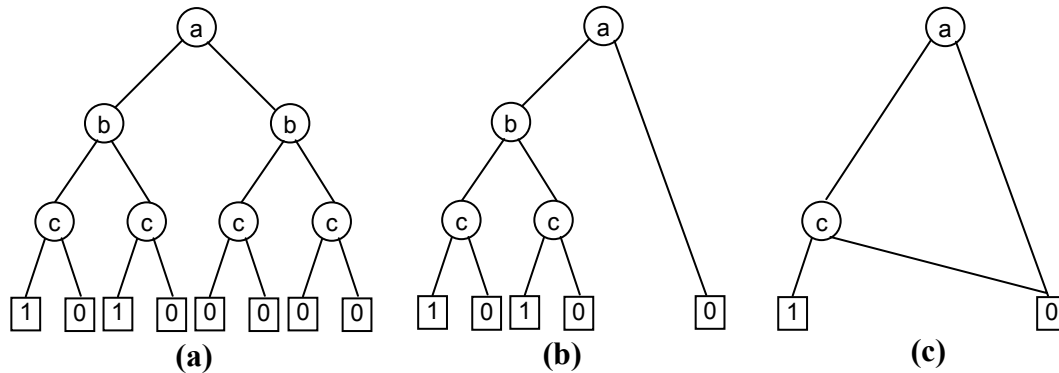
PENDAHULUAN

Berbagai metode kompresi citra telah dikembangkan dalam rangka mencapai rasio kompresi yang tinggi dengan tetap mempertahankan mutu. Secara umum kompresi citra terbagi menjadi dua, yaitu *lossless* dan *lossy*. Untuk kelompok *lossless* dikenal beberapa metode misalnya *Run Length Encoding* yang diterapkan pada file PCX, LZW pada file TIFF, Quadtree dan Pengkodean Huffman, dimana dengan metode ini seluruh data dipertahankan atau dengan kata lain tidak ada data yang hilang. Metode-metode untuk kompresi *lossless* biasanya mempunyai rasio kompresi yang rendah dan kebanyakan dikerjakan pada piksel secara linier, tidak memandang citra sebagai blok-blok kecil. Padahal salah satu karakteristik yang membedakan antara citra dan data lain adalah karakteristik spasialnya, artinya nilai antara piksel yang berdekatan secara spasial biasanya hampir sama. Karakteristik ini yang didekati dengan penggunaan OBDD.

Binary Decision Diagram (BDD) merupakan *directed acyclic graph* dengan dua tipe simpul yang berbeda, yaitu simpul terminal dan simpul non-

terminal. Simpul terminal mewakili nilai Boolean '0' dan '1' sedangkan simpul non-terminal menyatakan variabel dari fungsi yang akan direpresentasikan oleh BDD ini. Setiap simpul non-terminal mempunyai dua cabang keluaran yang ditandai dengan konstanta Boolean '1' pada satu cabang dan '0' pada cabang yang lain [2,3]. *Graph BDD* selalu diawali dengan satu simpul yang disebut sebagai *root* yang tidak mempunyai induk/cabang masukan. BDD memiliki dua sifat penting yaitu bisa diurutkan dan bisa disederhanakan [9]

Ordered BDD (OBDD) merupakan BDD yang setiap variabelnya muncul tidak lebih dari satu kali pada setiap jalurnya dan selalu muncul dengan urutan yang sama. OBDD pada mulanya digunakan untuk mengurangi kebutuhan kapasitas penyimpanan dan perhitungan yang dibutuhkan untuk menguji kebenaran suatu untai digital. OBDD akan membuang bagian-bagian *sub-function* yang kembar (*redundant*) yang terdapat pada fungsi Boolean. Gambar 1 menunjukkan contoh representasi sebuah fungsi biner ($a \bullet c$), berturut-turut dengan binary tree, quadtree dan OBDD [2].



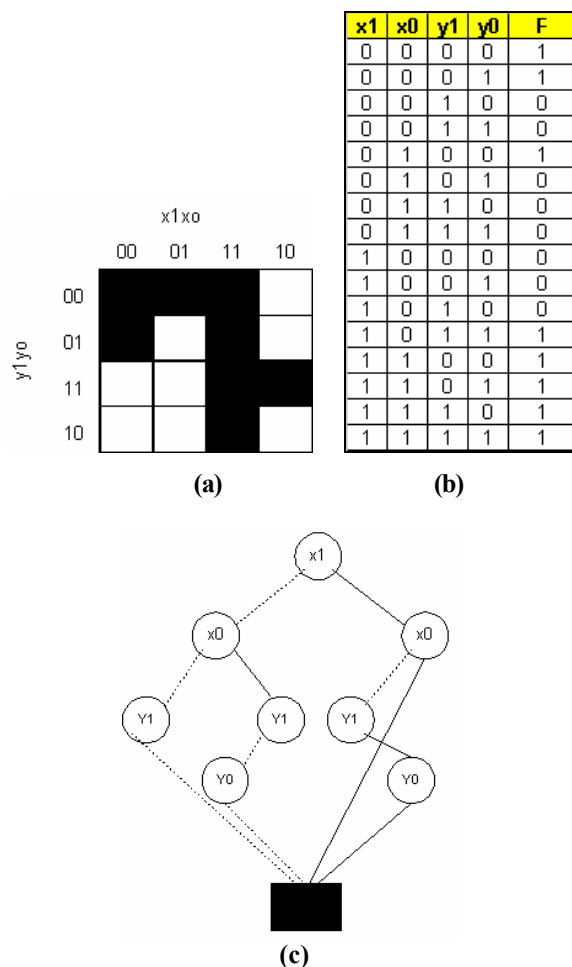
Gambar 1. Representasi fungsi biner $(a \cdot c)$ (a) dengan binary tree (b) dengan binary Quadtree (c) dengan OBDD

OBDD dan Pengkodean Huffman juga telah digunakan sebagai algoritma kompresi citra [4,5,10]. Untuk kompresi citra, OBDD diterapkan pada blok-blok kecil dari citra yang telah dibagi-bagi. Berbagai penelitian yang dilakukan selama ini hanya untuk citra *grayscale*.

REPRESENTASI OBDD UNTUK CITRA BITMAP

OBDD untuk kompresi citra adalah teknik kompresi yang diperkenalkan oleh Starkey dan Bryant [8], serta Lursinsap, Kanchanasut dan Siriboon [7]. Sebuah citra hitam-putih (*binary image*) dapat dinyatakan sebagai fungsi biner dimana citra tersebut dipandang sebagai sebuah *karnaugh-map* (K-Map). Dengan demikian sebuah piksel yang dapat dinyatakan sebagai sebuah fungsi jika sepanjang koordinat x dan koordinat y diberi variabel biner tertentu (Gambar 2a). Jika secara konsisten diberi cabang (kiri atau kanan) berdasarkan nilai bit variabel maka OBDD dapat disusun untuk suatu citra. Nilai koordinat menggunakan penomoran *big-endian* yang lebar bitnya tergantung dari ukuran blok yang akan diproses.

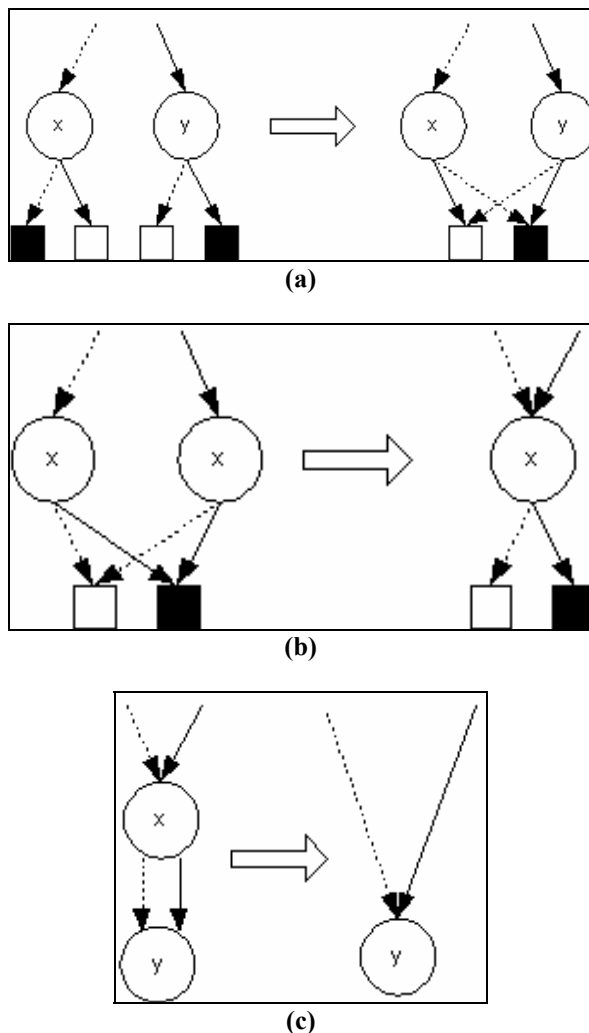
OBDD pada makalah ini mengacu pada OBDD yang diusulkan oleh Lursinsap, Kanchanasut and Siribon [7] dan dibangun dengan urutan variabel tertentu. OBDD yang sederhana didapatkan dengan menerapkan algoritma penyederhanaan dan aturan transformasi. Sebagai contoh, citra pada gambar 2(a) jika dinyatakan dengan fungsi *sum of product* (SOP) dapat ditulis menjadi $f = x1' \cdot y1' \cdot y0' + x1 \cdot x0 + x1' \cdot x0' \cdot y1' + x1 \cdot y1 \cdot y0$, dimana hasil dari OBDD ditunjukkan pada gambar 2(c).



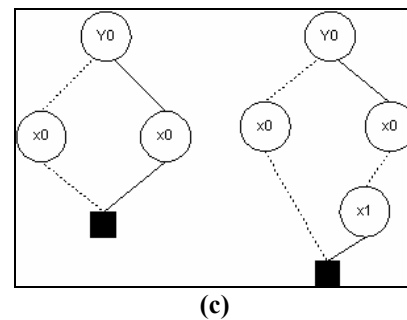
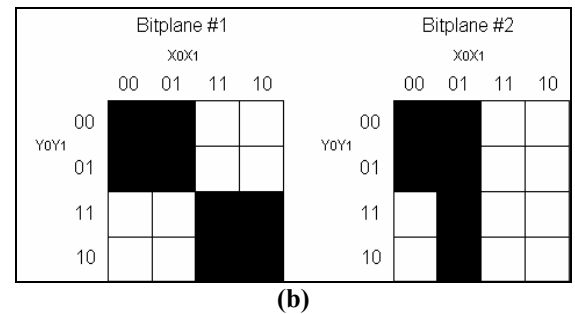
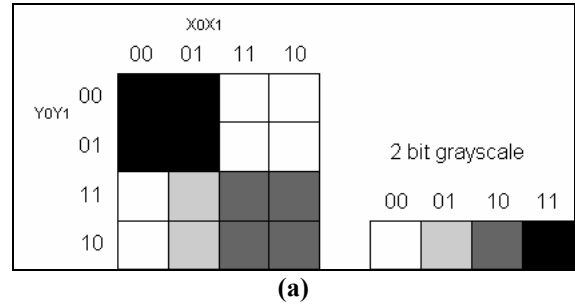
Gambar 2. (a) citra sederhana (4 x 4); (b) tabel kebenaran dari citra untuk menyusun OBDD (c) representasi OBDD untuk citra (a)

Ada tiga aturan penting untuk membuang simpul kembar yang muncul pada saat membangun OBDD, yaitu [1,4]:

- a) Membuang terminal yang kembar: Simpul terminal yang mempunyai nilai sama dapat diwakili dengan sebuah simpul saja dan semua cabang yang mengarah ke simpul terminal tersebut diarahkan ke simpul yang mewakili, seperti dicontohkan dalam gambar 3a.
- 2) Membuang Non-Terminal yang kembar: Jika simpul non-terminal u dan v menyatakan variabel yang sama $\text{var}(u) = \text{var}(v)$ dan keduanya mempunyai anak yang sama baik pada cabang '0' maupun cabang '1', maka salah satu simpul tersebut bisa dihilangkan dan input dari keduanya diarahkan ke simpul yang mewakili, seperti dicontohkan dalam gambar 3b.
- 3) Membuang Redundant Test: Jika kedua cabang yang keluar dari simpul non-terminal menuju ke simpul yang sama, maka simpul tersebut bisa dihilangkan karena evaluasi fungsi boolean yang dinyatakan tidak tergantung pada simpul ini (gambar 3c).



Gambar 3. Aturan untuk Membuang Simpul Kembar



Gambar 4. (a) Citra derajat keabuan 2-bit, (b) Citra biner untuk tiap bit-plane (a), (c) OBDD bagi tiap bit-plane.

Analisis Running Time Complexity

a. Kondisi terbaik (*Best Case*)

Akan tercapai bila semua piksel dalam blok '0' atau '1'. Pada kasus ini tidak tergantung pemilihan urutan variabel yang digunakan, karena akan didapatkan hasil yang selalu sama. Pada kondisi ini OBDD akan memiliki:

- Tinggi level: 1
- Jumlah Simpul non terminal: 1
- Big-Oh time complexity $O(1)$

b. *Worst Case*

Kondisi terburuk akan tercapai jika piksel dalam blok membentuk pola seperti papan catur. Secara nyata, citra yang mirip ini adalah citra dengan perubahan intensitas yang sangat cepat tiap pikselnya. Pada kondisi ini OBDD akan memiliki:

- Tinggi level: $\log_2 [M+N]$
 - Jumlah Simpul non terminal: $2^N - 1$
 - Big-Oh time complexity: $\Theta(2^{(\lceil \log_2 M \rceil + \lceil \log_2 N \rceil + 1)})$
- Dimana M dan N adalah ukuran blok.

YUV

YUV, dikenal juga sebagai Y'CbCr dan YpbPr, adalah sebuah *color space* dengan Y merupakan komponen *luminance* (*brightness*) dan U dan V adalah komponen *chrominance* (warna). YUV biasanya digunakan pada aplikasi video. Sinyal YUV diciptakan dari sumber RGB (*red, green, blue*), dengan diberi bobot tertentu, nilai R, G dan B ditambahkan untuk menghasilkan sinyal Y, yang menyatakan terang-gelap secara keseluruhan, atau *luminance* dari titik tersebut. Sinyal U lalu dibentuk dengan mengurangkan sinyal *blue* dari RGB dengan Y, dan V dengan mengurangkan *red*.

Pada makalah ini diterapkan kompresi citra metode OBDD pada citra berwarna. Pada mode *lossless*, OBDD akan bekerja pada tiap bit plane seperti pada penelitian sebelumnya, sedangkan pada mode *lossy*, konversi dari RGB ke YUV akan dilakukan sebelum proses dengan OBDD.

Ada berbagai rumus konversi dari RGB ke YUV yang dapat digunakan, namun dalam makalah ini digunakan rumus sebagai berikut:

$$Y = (0.256788 * R + 0.504129 * G + 0.097906 * B) + 16 \quad (1)$$

$$U = (-0.148223 * R - 0.290993 * G + 0.439216 * B) + 128 \quad (2)$$

$$V = (0.439216 * R - 0.367788 * G - 0.071427 * B) + 128 \quad (3)$$

Dan pengubah balik dari YUV ke RGB dilakukan dengan rumus:

$$R = 1.164383 * Y * 0.9 + 1.596027 * (V - 128) \quad (4)$$

$$G = 1.164383 * Y * 0.9 - 0.391762 * (U - 128) - 0.812968 * (V - 128) \quad (5)$$

$$B = 1.164383 * Y * 0.9 + 2.017232 * (U - 128) \quad (6)$$

HASIL UJI COBA

Untuk menguji kompresi citra berwarna dengan OBDD, beberapa citra standar untuk kompresi digunakan. Citra yang digunakan untuk percobaan adalah citra 24 bit dengan ukuran 256x256 dan 512x512 piksel. Citra-citra ini diklasifikasikan ke dalam beberapa kelompok yang dinamai natural (pemandangan dan hewan), teks, tekstur, dan citra buatan. Setiap kelompok terdiri dari 7 hingga 8 citra yang berbeda masing-masing dua ukuran (256x256 dan 512x512 piksel). Gambar 5 menunjukkan sebagian dari citra yang digunakan pada percobaan, sedangkan Tabel 1 menunjukkan statistik ukuran dari citra yang digunakan.



Gambar 5. Sampel Citra yang Digunakan dan Klasifikasinya

Tabel 1. Statistik Ukuran Citra yang Digunakan

Kelompok Citra	Jumlah Citra yang Digunakan	
	256X256 Piksel	512X512 Piksel
Natural	8 buah	8 buah
Teks	7 buah	7 buah
Tekstur	8 buah	8 buah
Buatan	7 buah	7 buah

Dari hasil percobaan (Tabel 2) terlihat bahwa OBDD mencapai hasil terbaiknya jika digunakan untuk mengompresi citra dengan warna yang berupa blok-blok seperti citra buatan dan citra teks. Tabel tersebut juga menunjukkan bahwa rasio kompresi untuk citra buatan terbaik di antara citra kategori lain. Hal ini disebabkan setiap bit-plane dari citra tekstur atau natural bisa dikatakan random total, sehingga

tidak mempunyai pola yang dapat dikompresi dengan OBDD. Dibandingkan dengan *Lossless* JPEG, OBDD lebih buruk untuk semua kategori.

Tabel 2. Perbandingan rasio kompresi antara OBDD dan *Lossless* JPEG

Kategori	OBDD	OBDD	OBDD	JLS
	16 x 16	32 X 32	64 X 64	
<i>Natural</i>	1.202	1.230	1.214	1.726
Buatan	29.983	23.850	63.255	172.559
Tekstur	1.308	1.290	1.298	1.754
Teks	5.723	5.293	6.317	15.171

Tabel 3. Rasio Kompresi OBDD dengan YUV 4:1:1, Kompresi OBDD dengan YUV 4:2:2 dan JPEG2000

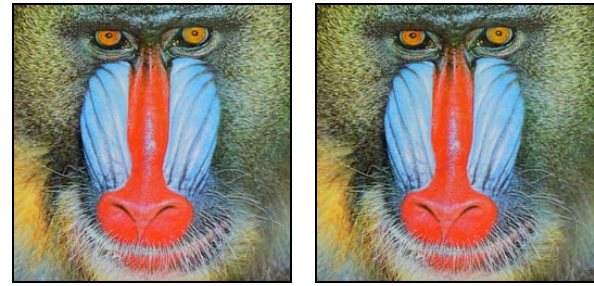
Kategori	YUV 4:1:1			YUV 4:2:2			JP2K
	16 x 16	32 X 32	64 X 64	16 x 16	32 X 32	64 X 64	
<i>Natural</i>	2.477	2.543	2.503	1.903	1.955	1.926	2.870
Buatan	42.356	37.846	92.040	33.716	28.191	71.731	88.423
Tekstur	2.560	2.528	2.546	1.988	1.962	1.978	2.890
Teks	11.440	11.938	13.483	9.738	10.488	12.008	10.962

Tabel 4. SNR (dB) Kompresi OBDD dengan YUV 4:1:1, Kompresi OBDD dengan YUV 4:2:2, JPEG2000

Kategori	OBDD		JP2K
	YUV 4:1:1	YUV 4:2:2	
<i>Natural</i>	61.8	69.7	67.5
Buatan	26.1	30.7	104.3
Tekstur	62.3	60.4	57.2
Teks	33.6	34.2	115.8

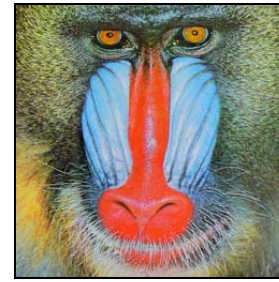
Dari hasil pengujian pada Tabel 3 dan Tabel 4 dapat dilihat bahwa penggunaan YUV 4:1:1 akan menghasilkan kecepatan dan rasio kompresi yang lebih baik dibandingkan dengan YUV 4:2:2, tetapi penggunaan YUV 4:1:1 menghasilkan SNR yang lebih buruk dibandingkan dengan penggunaan YUV 4:2:2 (selisih 7,9 dB pada kategori *natural*). Hal tersebut karena YUV 4:1:1 dapat mewakili citra cukup dengan jumlah bit $\frac{1}{2}$ dari citra asal sedangkan YUV 4:2:2 dengan $\frac{2}{3}$ jumlah bit citra asal. Penggunaan bit yang lebih sedikit akan menghasilkan rasio kompresi dan waktu kompresi serta waktu dekompresi yang lebih baik, tetapi hal ini juga akan mengorbankan kualitas citra pada saat citra direka ulang.

Pengaruh ukuran blok OBDD sama seperti pada kompresi OBDD biasa, yaitu rasio kompresi terbaik dengan blok berukuran 64 x 64 dan waktu kompresi tercepat dengan blok berukuran 16 x 16. Ukuran blok OBDD tidak mempengaruhi kualitas citra karena sifat dari kompresi OBDD sendiri yang *lossless*.

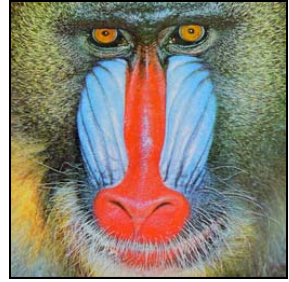


a) Citra asli

b) Citra hasil kompresi OBDD dan YUV 4:1:1



c) Citra hasil kompresi OBDD dan YUV 4:2:2



d) Citra hasil kompresi JPEG2000

Gambar 6. Contoh hasil kompresi OBDD dan JPEG2000

Dari hasil yang didapat maka penggunaan YUV 4:1:1 lebih disukai karena menghasilkan rasio kompresi yang lebih baik (30% lebih baik dibanding YUV 4:2:2), waktu kompresi yang lebih cepat (30% lebih cepat dibanding YUV 4:2:2), waktu dekompresi yang lebih cepat (30% lebih cepat dibanding YUV 4:2:2) dengan SNR yang masih bisa diterima (lebih besar dari 20dB) [11].

Pengujian dan analisa secara subyektif dilakukan dengan menilai kualitas citra oleh responden. Hal ini perlu dilakukan karena sebenarnya kompresi *lossy* hanya diperbolehkan selama *noise* tidak terlihat oleh mata manusia. Hasil yang didapat digunakan untuk melengkapi hasil pengujian dan analisa secara obyektif. Gambar 6 menunjukkan contoh hasil kompresi OBDD dan YUV 4:1:1 dibanding dengan JPEG 2000.

Tabel 5. Kualitas Citra Secara Subyektif Kompresi OBDD dengan YUV 4:1:1, Kompresi OBDD dengan YUV 4:2:2, dan JPEG2000

Kategori	OBDD		JP2000
	YUV 411	YUV 422	
<i>Natural</i>	9.3	9.3	9.4
Buatan	8.4	9.0	9.4
Tekstur	9.3	9.4	9.4
Teks	8.6	9.0	9.3

Dari hasil perbandingan (Tabel 5) dapat dilihat bahwa pada kategori natural dan tekstur kualitas citra untuk ketiga metode kompresi setara. Sedangkan pada kategori buatan dan teks kualitas citra berurutan dari yang paling buruk adalah kompresi OBDD dengan YUV 4:1:1, kompresi OBDD dengan YUV 4:2:2, dan JPEG2000.

Dari hasil yang didapat terlihat bahwa sekalipun SNR sebuah citra berbeda cukup jauh tapi pada manusia hal itu bisa saja tak bermasalah. Hal ini disebabkan oleh kemampuan mata manusia yang terbatas dan perhitungan SNR yang hanya melihat *noise* pada tiap *piksel* padahal antara *piksel* yang bertetangga dapat saling mempengaruhi.

Dari hasil yang diperoleh dapat ditarik kesimpulan bahwa kompresi OBDD dengan YUV 4:1:1 lebih dipilih daripada kompresi OBDD dengan YUV 4:2:2 karena dengan kualitas citra secara subjektif yang memadai dan dapat diperoleh rasio kompresi yang lebih baik. Pertimbangan yang lain adalah bahwa pada kategori natural, yang lebih banyak dijumpai dalam kehidupan sehari-hari, kompresi OBDD dengan YUV 4:1:1 menghasilkan kualitas citra subjektif yang setara dengan kompresi OBDD dengan YUV 4:2:2.

Perbandingan kualitas citra subjektif antara kompresi OBDD dengan YUV 4:1:1, kompresi OBDD dengan YUV 4:2:2, dan JPEG2000 memberikan hasil bahwa, pada kategori natural dan tekstur kompresi OBDD dengan YUV 4:1:1 dan kompresi OBDD dengan YUV 4:2:2 setara JPEG2000; dan pada kategori buatan dan teks kompresi OBDD dengan YUV 4:1:1, kompresi OBDD dengan YUV 4:2:2 lebih buruk daripada JPEG2000.

Pada makalah ini tidak dibandingkan waktu proses antara OBDD dengan JPEG2000 karena digunakan lingkungan pemrograman yang berbeda antara aplikasi-aplikasi ini. Namun secara teoritis, Lossless JPEG menggunakan perhitungan-perhitungan yang jauh lebih rumit dibandingkan dengan OBDD dan oleh karena alasan yang sama JPEG2000 dapat menghasilkan faktor kompresi yang lebih baik.

KESIMPULAN

Kompresi OBDD untuk citra berwarna paling sesuai menggunakan blok berukuran 64 x 64 untuk menghasilkan rasio kompresi yang terbaik atau 16 x 16 untuk menghasilkan proses kompresi yang tercepat. Kompresi RGB→YUV→OBDD menggunakan YUV 4:1:1 untuk menghasilkan rasio kompresi terbaik dalam waktu kompresi yang tercepat dengan kualitas citra yang memadai.

Kompresi OBDD memiliki rasio kompresi yang lebih buruk dibandingkan Lossless JPEG. Hanya pada kategori *natural* dan *texture* kompresi OBDD

dapat mendekati Lossless JPEG. Kompresi RGB→YUV→OBDD dengan YUV 4:1:1 dan ukuran blok 64 x 64 memiliki SNR yang lebih buruk untuk citra kategori *natural*, *synthetic*, *text* dan lebih baik untuk citra kategori *texture* dibandingkan JPEG2000 dengan rasio kompresi yang setara.

Kompresi RGB→YUV→OBDD dengan YUV 4:1:1 dan ukuran blok 64 x 64 memiliki rasio kompresi yang lebih buruk dibandingkan JPEG2000 dengan SNR yang setara. Hanya pada kategori *texture* rasio kompresi RGB→YUV→OBDD dapat mendekati JPEG2000.

Kompresi RGB→YUV→OBDD dengan YUV 4:1:1 dan YUV 4:2:2 memiliki kualitas citra secara visual yang setara untuk citra kategori *natural* dan *texture*; dan lebih buruk untuk citra kategori *synthetic* dan *text* dibandingkan JPEG2000 dengan rasio kompresi yang setara.

DAFTAR PUSTAKA

1. Akers, S. B., "Binary Decision Diagrams", *IEEE Transaction on Computers*, Vol. C-27 No.6. 1978.
2. Bryant, R. E., "Graph Based Algorithms for Boolean Function Manipulation", *IEEE Transaction on Computers*, Vol. C-35 No.8. 1986.
3. Bryant, R. E., "Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams", *ACM Computing Surveys*, Vol. 24, No.3. 1992.
4. Handoko, "Kompresi Citra Lossless dengan Menggunakan Quadtree-OBDD", *Majalah IPTEK, Institut Teknologi Surabaya*, Surabaya, 2004.
5. Handoko, and Utomo V.G., *Kompresi Citra Berwarna dengan OBDD*, Skripsi, Fakultas Teknik Universitas Kristen Satya Wacana, Salatiga. 2004.
6. Kanchanasut, K. and Handoko, *Hybrid Image Representation Using Quadtree and OBDD*, Master Thesis, Department of Computer Science, Asian Institute of Technology, Bangkok. 2002.
7. Lursinsap, C., Kanchanasut, K., Siriboon, T., *Basic Binary Decision Diagram Operations for Image Processing*, Proc. 3rd Asian Computing Science Conference, Springer-Verlag, Lecture Notes in Computer Science 1345, 1997, pp. 368-370.
8. Starkey, M., and Bryant, R., *Using Ordered Binary-Decision Diagrams for Compressing Images and Image Sequences*, Department of Computer Science, Carnegie Mellon University, Pittsburgh, 1995.
9. Townsend, W., and Thornton, A., *Partial Binary Decision Diagrams*, Mississippi State University,

- (<http://cs.engr.uky.edu/~lewis/papers/merge-alg.pdf>). 1999.
10. Witwiyaruj, K., *Design and Implementation of A Coding Scheme for Image Compression Using Ordered Binary-Decision Diagram*, Master Thesis, Department of Computer Science, Asian Institute of Technology, Bangkok. 2000.
 11. _____, Measuring Distortion in Lossy Compression Systems, www.wmin.ac.uk.